



EBook Gratuito

APPENDIMENTO

pyqt4

Free unaffiliated eBook created from
Stack Overflow contributors.

#pyqt4

Sommario

Di.....	1
Capitolo 1: Iniziare con pyqt4.....	2
Osservazioni.....	2
Examples.....	2
Installazione o configurazione.....	2
Programma base Hello World.....	2
Capitolo 2: Ciao programma mondiale.....	4
introduzione.....	4
Osservazioni.....	4
Examples.....	4
Ciao programma mondiale.....	4
Capitolo 3: QMessageBox.....	6
introduzione.....	6
Examples.....	6
Utilizzo di base: Hello World.....	6
Capitolo 4: Segnali e slot.....	7
introduzione.....	7
Osservazioni.....	7
Examples.....	7
Un esempio di utilizzo di segnali e slot.....	7
Capitolo 5: Widget di base: QLabel.....	9
introduzione.....	9
Osservazioni.....	9
Examples.....	9
Esempio di QLabel con testo, collegamento ipertestuale e immagine.....	9
Titoli di coda.....	12

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [pyqt4](#)

It is an unofficial and free pyqt4 ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official pyqt4.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capitolo 1: Iniziare con pyqt4

Osservazioni

Questa sezione fornisce una panoramica di cosa sia `PyQt4` e perché uno sviluppatore potrebbe volerlo utilizzare.

PyQt è un toolkit per i widget della GUI. È un'interfaccia Python per Qt, una delle più potenti e diffuse librerie GUI multiplatforma. PyQt è una miscela del linguaggio di programmazione Python e della libreria Qt. Questo tutorial introduttivo ti aiuterà a creare applicazioni grafiche con l'aiuto di PyQt.

Dovrebbe anche menzionare qualsiasi argomento di grandi dimensioni all'interno di `PyQt4` e collegarsi agli argomenti correlati. Poiché la Documentazione per `PyQt4` è nuova, potrebbe essere necessario creare versioni iniziali di tali argomenti correlati.

Examples

Installazione o configurazione

Istruzioni dettagliate su come installare o installare `pyqt4`.

1. [Collegamento di Windows](#)
2. Mac [Link](#)
3. [Link Linux](#)

Se si desidera installare la versione specifica per la propria versione python di sistema e la configurazione del sistema (32 bit o 64 bit), andare a questo [link](#) e scaricare e installare il pacchetto. È possibile installare il file `.whl` semplicemente andando al prompt dei comandi `pip install PyQt4-4.11.4-cp34-none-win_amd64.whl`.

Vai a testa Installa il software e inizia a costruire una GUI impressionante !!

Programma base Hello World

```
import sys
from PyQt4 import QtGui

def window():
    app = QtGui.QApplication(sys.argv)
    w = QtGui.QWidget()
    b = QtGui.QLabel(w)
    b.setText("<h1>Welcome to PyQt4 SO Documentation!</h1>")
    w.setGeometry(100,100,550,65)
    b.move(50,20)
    w.setWindowTitle("PyQt4 Hello World Demo")
    w.show()
    sys.exit(app.exec_())
```

```
if __name__ == '__main__':  
    window()
```

Leggi Iniziare con pyqt4 online: <https://riptutorial.com/it/pyqt4/topic/8665/iniziare-con-pyqt4>

Capitolo 2: Ciao programma mondiale

introduzione

È necessario conoscere le basi del linguaggio di programmazione Python prima di iniziare con PyQt.

PyQt è un toolkit per i widget della GUI. È un'interfaccia Python per Qt, una delle più potenti e diffuse librerie GUI multiplatforma. PyQt è una miscela del linguaggio di programmazione Python e della libreria Qt.

Ecco un programma Hello World per iniziare.

Osservazioni

La creazione di una semplice applicazione GUI mediante PyQt prevede i seguenti passaggi:

1. Importa il modulo QtGui
2. Creare un oggetto applicazione.
3. Un oggetto QWidget crea una finestra di livello superiore. Aggiungi l'oggetto QLabel in esso.
4. Imposta la didascalia dell'etichetta come "ciao mondo".
5. Definire la dimensione e la posizione della finestra con il metodo setGeometry ().
6. Inserisci il mainloop dell'applicazione dal metodo app.exec_ ().

Examples

Ciao programma mondiale

```
import sys
from PyQt4 import QtGui

def window():
    app = QtGui.QApplication(sys.argv)
    w = QtGui.QWidget()
    b = QtGui.QLabel(w)
    b.setText("Hello World!")
    w.setGeometry(100,100,200,50)
    b.move(50,20)
    w.setWindowTitle("PyQt")
    w.show()
    sys.exit(app.exec_())

if __name__ == '__main__':
    window()
```

Leggi Ciao programma mondiale online: <https://riptutorial.com/it/pyqt4/topic/9247/ciao-programma-mondiale>

Capitolo 3: QMessageBox

introduzione

`QMessageBox` è il modo più semplice per fornire (o chiedere) informazioni all'utente (o da). È una finestra di dialogo modale che eredita la classe `QDialog`. Ha anche quattro funzioni statiche di convenienza: `information`, `question`, `warning` e `critical`.

Examples

Utilizzo di base: Hello World

```
app = QApplication( sys.argv )
box = QMessageBox()

# Window Title
box.setWindowTitle( "Hello World." )

# Icon: Information, Warning, Question, Critical
box.setIcon( QMessageBox.Information )

# Short version of the information
box.setText( "Hello World!" )

# Informative text
box.setInformativeText( "Hello World! We are using Qt to display this beautiful dialog to you." )

# Show the messagebox as a modal dialog
box.exec_()

return 0
```

Leggi QMessageBox online: <https://riptutorial.com/it/pyqt4/topic/9806/qmessagebox>

Capitolo 4: Segnali e slot

introduzione

Funzioni o metodi vengono eseguiti in risposta alle azioni dell'utente, come `clicking` su un pulsante, selezionare un elemento da una raccolta o un clic del mouse ecc., Chiamati **events** .

Ogni widget PyQt, derivato dalla classe `QObject` , è progettato per emettere il **segnale** in risposta a uno o più eventi. Il segnale da solo non esegue alcuna azione. Invece, è **collegato** a uno **slot** .

Osservazioni

Nell'esempio seguente, due oggetti `QPushButton` (`b1` e `b2`) vengono aggiunti nella finestra `QDialog` . Vogliamo chiamare le funzioni `b1_clicked()` e `b2_clicked()` facendo clic rispettivamente su `b1` e `b2` .

Quando si fa clic su `b1` , il segnale `clicked()` è collegato alla funzione `b1_clicked()`

```
b1.clicked.connect(b1_clicked())
```

Quando viene cliccato `b2` , il segnale `clicked()` è collegato alla funzione `b2_clicked()`

```
QObject.connect(b2, SIGNAL("clicked()"), b2_clicked)
```

I widget usati per costruire l'interfaccia della GUI fungono da fonte di tali eventi.

Ogni widget PyQt, derivato dalla classe `QObject` , è progettato per emettere il **segnale** in risposta a uno o più eventi. Il segnale da solo non esegue alcuna azione. Invece, è **collegato** a uno **slot** . Lo slot può essere una **funzione Python richiamabile** .

Examples

Un esempio di utilizzo di segnali e slot

```
import sys
from PyQt4.QtCore import *
from PyQt4.QtGui import *

def window():
    app = QApplication(sys.argv)
    win = QDialog()
    b1 = QPushButton(win)
    b1.setText("Button1")
    b1.move(50,20)
    b1.clicked.connect(b1_clicked)

    b2 = QPushButton(win)
    b2.setText("Button2")
    b2.move(50,50)
```

```
QObject.connect(b2, SIGNAL("clicked()"), b2_clicked)

win.setGeometry(100, 100, 200, 100)
win.setWindowTitle("PyQt")
win.show()
sys.exit(app.exec_())

def b1_clicked():
    print ("Button 1 clicked")

def b2_clicked():
    print ("Button 2 clicked")

if __name__ == '__main__':
    window()
```

L'esempio seguente produrrà due pulsanti, se si fa clic su di essi, verrà richiamata la funzione `b1_clicked` o `b2_clicked`.

Leggi Segnali e slot online: <https://riptutorial.com/it/pyqt4/topic/9270/segnali-e-slot>

Capitolo 5: Widget di base: QLabel

introduzione

`QLabel` è utilizzato per la visualizzazione di testo o un'immagine. Non viene fornita alcuna funzionalità di interazione dell'utente. L'aspetto visivo dell'etichetta può essere configurato in vari modi e può essere usato per specificare un tasto mnemonico di messa a fuoco per un altro widget.

Osservazioni

Un oggetto `QLabel` funge da segnaposto per visualizzare **testo** , **immagine** o **GIF animati** *non modificabili* . Può anche essere usato come chiave mnemonica per altri widget.

In questo esempio, `QLabel` oggetti `QLabel 12` e `14` hanno la didascalia che contiene il collegamento ipertestuale. `setOpenExternalLinks` per `12` è impostato su **true** . Quindi, se questa etichetta viene **cliccata** , l'URL associato si aprirà nel browser. `linkHovered signal` of `14` è connesso alla funzione `linkHovered hovered()` . Quindi, ogni volta che il mouse passa su di esso, la funzione verrà eseguita.

L'oggetto `QPixmap` prepara l'immagine fuori schermo dal file `python.jpg` . Viene visualizzato come etichetta `13` utilizzando il metodo `setPixmap()`

Anche testo semplice , **collegamento ipertestuale** o **rich text** possono essere visualizzati in `QLabel` .

In `QLabel` maggior parte dei tag HTML di base. Per esempio: `h1` , `h2` , `h3` , `font` , `span` , ecc

[Puoi visitare qui per l'elenco dei sottoinsiemi HTML supportati.](#)

Examples

Esempio di QLabel con testo, collegamento ipertestuale e immagine

Di seguito è riportato l'esempio di `QLabel` che mostra l'uso di testi, immagini e collegamenti ipertestuali.

```
import sys
from PyQt4.QtCore import *
from PyQt4.QtGui import *

def window():
    app = QApplication(sys.argv)
    win = QWidget()

    l1 = QLabel()
    l2 = QLabel()
```

```

l3 = QLabel()
l4 = QLabel()

l1.setText("<h1>Hello World</h1>")
l4.setText("<b>Hello Stack OverFlow</b>")
l2.setText("<font color='red'>Welcome To Stack Overflow Documentation</font>")

l1.setAlignment(Qt.AlignCenter)
l3.setAlignment(Qt.AlignCenter)
l4.setAlignment(Qt.AlignRight)
l3.setPixmap(QPixmap("python.jpg"))

vbox = QVBoxLayout()
vbox.addWidget(l1)
vbox.addStretch()
vbox.addWidget(l2)
vbox.addStretch()
vbox.addWidget(l3)
vbox.addStretch()
vbox.addWidget(l4)

l1.setOpenExternalLinks(True)
l4.linkActivated.connect(clicked)
l2.linkHovered.connect(hovered)
l1.setTextInteractionFlags(Qt.TextSelectableByMouse)
win.setLayout(vbox)

win.setWindowTitle("PyQt4 QLabel Demo")
win.show()
sys.exit(app.exec_())

def hovered():
    print ("Come'On Click ME")
def clicked():
    print ("You Clicked Me")

if __name__ == '__main__':
    window()

```

Il codice restituisce il seguente risultato:



Leggi Widget di base: QLabel online: <https://riptutorial.com/it/pyqt4/topic/9312/widget-di-base--qlabel>

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con pyqt4	Ahmad Taha , Community , learncode
2	Ciao programma mondiale	Ahmad Taha
3	QMessageBox	Marcus
4	Segnali e slot	Ahmad Taha
5	Widget di base: QLabel	Ahmad Taha